REMARKS

Claims 1-20 are pending in the application. Claims 1, 4-5, 7-8, 11-15, and 17-20 have been amended. Accordingly, Claims 1-20 remain pending in the application.

35 U.S.C. § 102 and § 103 Rejections

Claims 1, 4-8, 11-15, and 17-20 were rejected under 35 U.S.C. 102(e) as being anticipated by Beardsley et al. (U.S. Patent Application Publication No. 2003/0131285). Claims 2, 3, 9, 10, and 16 were rejected under 35 U.S.C. 103(a) as being unpatentable over Beardsley in view of Topley (J2ME in a Nutshell).

1. Applicant respectfully submits that Beardsley fails to teach or suggest, "receiving requests at said server from said computing devices requesting said server to provide test programs to said computing devices" as recited by claim 1.

Beardsley teaches, "beginning at step 500, the test component 202 receives a test packet...from the product developer client 202...via the API 220" (Paragraph [0042]). Beardsley further teaches, "the test component 202 searches, via the autolab component 230, for an available client machine 212, 214 for performing the tests in the test packet. As further described below, an available client machine 212, 214 may be idle and awaiting a test packet, or may already be running the tasks in a test packet" (paragraph [0043]). In Beardsley, "If a client computer 212, 214 is available...the test component 202 checks to see if the computer is usable. That is, the autolab component 230 determines whether the client computer includes a group and application that meets the requirements of a pending test packet. If not, the process branches back...where a check is made for other idle client computers 212, 214" (paragraph [0047]). (Emphasis added) (see also paragraphs [0039] and [0046])

Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. M.P.E.P 2131;

Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co., 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). (Emphasis added)

While Beardsley teaches that the autolab 230 of the test component 202 searches for an available client machine 212, 214 to perform the tests and if it finds an available client computer it then checks to see if the available client is usable (otherwise another client must be used), Beardsley fails to teach "receiving **requests** at said server **from said computing devices requesting** said server to provide test programs to said computing devices" as recited by claim 1. In Beardsley, the autolab 230 of the test component 202 has the responsibility of searching for an available and usable client computer, instead of receiving requests from available client computers.

2. Additionally, Applicant submits that Beardsley fails to teach or suggest, "receiving requests at said server from said computing devices with respect to said execution of said test programs to determine a next test to execute at each of the corresponding computing devices, wherein each of said requests contains said respective unique identifier" as recited by claim 1.

The Examiner contends on page 2 of the pending Office Action that paragraphs [0033] and [0042]-[0044] of Beardsley teach the above-highlighted features of claim 1. Applicant respectfully disagrees. In paragraphs [0033] and [0047]-[0049], Beardsley teaches:

> [0033] Test conditions may be provided to the test component 202 in a number of different ways. In general, the test conditions are provided as tasks that a product developer client 204 would like to be performed in particular platform(s) and language(s). Hereinafter, for ease of discussion, a selected platform and language are referred to herein as a "group." In the embodiment shown in FIG. 2, each product developer client 204 provides a separate test packet 206, 208, 210 for each group on which the product developer wants a product tested. The separate test packet defines tasks that the product developer wants conducted on the product in that group. The number of test packets 206, 208, 210 generated is set by the product developer client 204, and, in the example shown, the product

developer client 204_1 provides L test packets, the product developer client 204_2 provides M test packets, and the product developer client 204_3 provides N test packets. A product developer client 204 may provide only one test packet, or may provide several test packets, depending upon the scope of the testing desired.

[0042] In any event, beginning at step 500, the test component 202 receives a test packet...from the product developer client...via the API 220. The received test packet is placed in a "pending" status file 602 (FIG. 6) in the database 222 at step 502.

[0043] At step 504, the autolab component 230 retrieves one of the pending test packets from the database 222. A determination is made if all tests have been run on the packet at step 506 (e.g., whether a record count is zero), and, if so, the process loops back and the next packet is retrieved at step 504. If not, then step 506 branches to step 508, where the test component 202 searches, via the autolab component 230, for an available client machine 212, 214 for performing the tests in the test packet. As further described below, an available client machine 212, 214 may be idle and awaiting a test packet, or may already be running the tasks in a test packet, but should be capable of (e.g., includes the proper groups and applications for) running the tasks of the test packet.

[0044] At step 510, the autolab component 230 configures the test packet to a personalized test package for the available client computer 212, 214. The test package is then placed, at step 512, in an "assigned" status file 604 (FIG. 6). The client machine 212, 214 is then assigned the test packet at step 514. Configuring and assigning the test package is described further in connection with FIG. 8, below. (Emphasis added)

Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). (Emphasis added)

While Beardsley teaches determining whether "all tests have been run on the packet" and "If not...the test component 202 searches, via the autolab component 230, for an available client machine 212, 214 for performing the tests in the test packet", Beardsley fails to teach, "**receiving requests at said server from said computing devices** with respect to said execution of said test programs **to determine** a next test to

execute at each of the corresponding computing devices, wherein each of said requests contains said respective unique identifier" as recited by claim 1.

In Beardsley, the autolab 230 of the test component 202 has the responsibility of searching for an available client computer, instead of receiving requests from available client computers. Specifically, Beardsley fails to teach the autolab 230 of the test component 202 "**receiving requests**" from the client machines 212, 214 "with respect to said execution of said test programs **to determine** a next test to execute at **each of the corresponding computing devices**, wherein each of said requests contains said respective unique identifier" (see above-highlighted features of claim 1).

Furthermore, as noted above and shown in FIG. 5, if all tests have not been run on the packet, in Beardsley the test component 202 searches for an available client to perform the test each time the process loops. However, in the present invention, the computing device that executed the previous test in the current test bundle executes the next test in the test bundle (see claim 1, "to determine a next test to execute at **each of the corresponding computing devices**", and see also claims 4, 5, and 7 which are highlighted below).

Accordingly, independent claim 1 is believed to patentably distinguish over Beardsley. Claims 2-7 are dependent upon claim 1 and are therefore believed to patentably distinguish over the cited references for at least the same reasons.

Likewise, claims 8 and 15 recite features similar to those highlighted above with regard to claim 1 and are therefore believed to patentably distinguish over Beardsley for at least the reasons given in the above paragraphs discussing claim 1. Claims 9-14 are dependent upon claim 8 and claims 16-20 are dependent upon claim 15, and are therefore believed to patentably distinguish over the cited references for at least the same reasons.

3. Due to the strict standard of anticipation (noted above), Applicant further contends that Beardsley fails to teach the features of various dependent claims. For instance:

4. Beardsley fails to teach, "said controlling the execution of said test programs comprises making a selection at said server, based on said respective unique identifier contained in said **requests**, of said next test to execute on each of said computing devices, and sending responses to said computing devices indicating said selection" as recited by claim 4. In accordance, claim 4 is believed to patentably distinguish over Beardsley.

5. Beardsley fails to teach, "wherein each of at least a subset of said test programs comprises a bundle of tests, the method further comprising: determining for **each** of the **requests** received at said server from said **computing devices** whether the received request is a **request** for the server to determine a new test bundle or a **request** for the server to determine a next test to be executed in a current test bundle" as recited by claim 5. In accordance, claim 5 is believed to patentably distinguish over Beardsley.

6. Beardsley fails to teach, "if the **request** is for a new test bundle, determining the new test bundle to provide to the **corresponding** computing device based on said respective unique identifier contained in said request and downloading the new test bundle from the server for execution by the **corresponding** computing device; and if the **request** is for a next test to be executed in a current test bundle, determining the next test based on said respective unique identifier contained in said request and providing information to the **corresponding** computing device of the next test to be executed in the current test bundle" as recited by claim 7. In accordance, claim 7 is believed to patentably distinguish over Beardsley.

7. Beardsley fails to teach, "wherein assigning said respective unique identifier comprises receiving an initial **request** from each of said computing devices to download one of said test programs, and assigning said respective unique identifier **in response to**

said initial **request**" as recited by claim 6. In accordance, claim 6 is believed to patentably distinguish over Beardsley.

8. Beardsley fails to teach, "said processor is configured to control said execution of said test programs in said suite based on said **received requests** and said **respective unique identifier** included therein, wherein said processor is configured to communicate responses to said received requests via said communication interface, each of said responses being addressed to a respective one of said computing devices that is associated with said respective unique identifier" as recited by claim 19. In accordance, claim 19 is believed to patentably distinguish over Beardsley.

CONCLUSION

Applicants submit the application is in condition for allowance, and an early notice to that effect is requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-62301/MJL.

Respectfully submitted,

Name Mario J. Lewin
Reg. No. 54,268
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin,
Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: 1/8/07